**INFORME: Desafío 1. Informática 2**

**Diana Lucia Baeza Ruiz, Sebastián Giraldo Álvarez**

1. **Análisis del problema.**

Necesitamos diseñar un sistema basado en Arduino que nos permita la adquisición de datos de una señal analógica. Para lo cual es importante asegurarnos de que cumpla las siguientes características:

* Iniciar y detener la adquisición de datos.
* Procesar los datos obtenidos para determinar las características de la señal tales como: frecuencia, amplitud y forma de onda.
* Visualizar los datos obtenidos en una pantalla LCD.

**Requerimientos:**

* Usar un pulsador para iniciar la adquisición de datos
* Usar un pulsador para detener la adquisición de datos
* Reanudar la adquisición de datos automáticamente.
* Usar un generador de señales de Tinkercad
* Implementar un código eficiente sin usar librerías, solo funciones propias.
* Para el código en C++ hacer uso de punteros, memoria dinámica y arreglos.
* A partir de los datos obtenidos en la adquisición se deben calcular las características de la señal análoga: Frecuencia medida en Hertz Hz, Amplitud medida en Voltios y Forma de onda
* Visualizar los datos obtenidos

1. **Consideraciones para la solución propuesta.**

Hacer el montaje en Tinkercad utilizando los siguientes componentes:

* 1 generador de señales
* 2 pulsadores
* 1 pantalla LCD (16x2)
* 3 resistencias de 1KΩ
* 1 Arduino UNO R3
* 1 potenciómetro de 10KΩ
* 1 placa de pruebas (protoboard)

Determinar las características de la seña análoga, mediante los datos obtenidos:

1. **Frecuencia**: La frecuencia (f) es la medida del número de ciclos o repeticiones de la onda por unidad de tiempo.

* Medir el número de repeticiones por segundo de la onda, por ejemplo, si una onda se repite diez veces por segundo, significa que tiene una frecuencia de diez ciclos por segundo (se expresa como frecuencia de 10 hercios o 10 Hz).
* Contar los cruces por cero de la señal (es decir, cuando la señal pasa de positivo a negativo). Esto se puede implementar mediante un algoritmo que captura los cambios de polaridad en un intervalo de tiempo.

1. **Amplitud**: La amplitud es la “altura” o distancia que tenga la forma de onda con respecto a la línea de cero voltios o bien entre pico positivo y negativo si la onda es de corriente alterna.

* Calcular el valor absoluto de cada uno de los datos obtenido y buscar entre ellos el máximo valor

1. **Identificación de la forma de la onda**: Las formas de onda se identifican comparando las características de la señal en un período de tiempo:

* La onda cuadrada tiene transiciones abruptas.
* La onda triangular presenta transiciones lineales.
* La onda senoidal tiene transiciones suaves.
* Si la entrada no corresponde con ninguna de las formas de onda anteriores, debe tener el estado de señal desconocida.

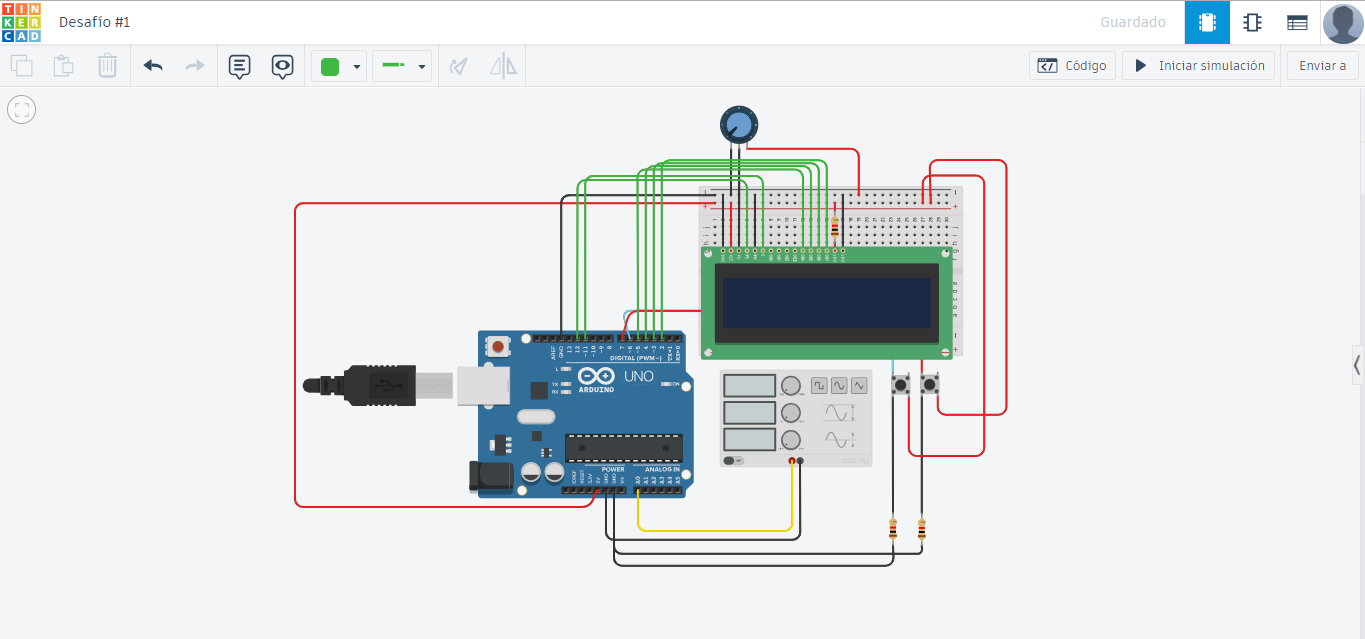
Se utilizarán punteros para gestionar los datos capturados de la señal y un arreglo dinámico para almacenar las lecturas de la señal en memoria, lo cual permite trabajar con un número variable de muestras.

1. **Diseño de la solución.**

Esquema de tareas a tener en cuenta para implementar la solución propuesta:

1. Se inicia con la configuración del sistema en Tinkercad usando los componentes indicados.
2. Se capturan los datos de la señal analógica desde el pin A0 y almacenándolo en un arreglo dinámico (puesto que no conocemos cuantos datos se van a recibir durante la adquisición).
3. Determinar los puntos de cruce por cero (cambios de signo) para calcular de manera eficiente la frecuencia de la señal.
4. Medir la amplitud identificando el valor máximo de la señal y convirtiéndolo a voltios.
5. Analizar las transiciones de la señal para identificar si es senoidal, cuadrada, triangular o ninguna.
6. Visualizar los resultados calculados en la pantalla LCD.
7. **Desarrollo de la solución**

**Configuración en Tinkercad**

****

**Conexiones**

* Entre el LCD y Arduino UNO

|  |  |
| --- | --- |
| **Pantalla LCD 16x2** | **Arduino UNO** |
| GND (Tierra) | GND |
| VCC (Potencia) | 5V |
| VD (Contraste) | Wiper (potenciómetro) – GND |
| RS (Selección de registro) | D12 |
| RW (Lectura/Escritura) | GND |
| E (Activar) | D11 |
| DB4 | D5 |
| DB5 | D4 |
| DB6 | D3 |
| DB7 | D2 |
| LED de ánodo | Resistencia de 1KΩ – 5V |
| LED de cátodo | GND |

* Entre el Pulsador 1 y Arduino UNO

|  |  |
| --- | --- |
| **Pulsador 1** | **Arduino UNO** |
| Terminal 1a | Resistencia 1KΩ – GND |
| Termina 2a | 5V |
| Termina 1b | D6 |

* Entre el Pulsador 2 y Arduino Uno

|  |  |
| --- | --- |
| **Pulsador 2** | **Arduino UNO** |
| Terminal 1a | Resistencia 1KΩ – GND |
| Termina 2a | 5V |
| Termina 1b | D7 |

* Generador de señales y Arduino UNO

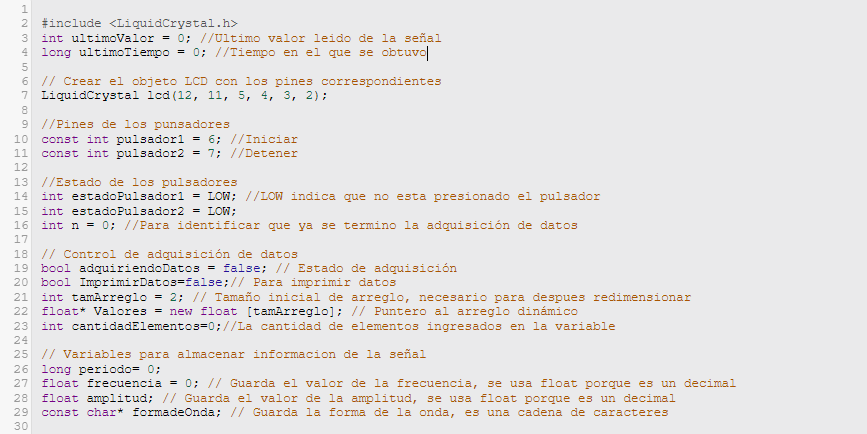
|  |  |
| --- | --- |
| **Generador de señales** | **Arduino UNO** |
| Positivo | A0 |
| Negativo | GND |

* Potenciómetro y LCD 16x2

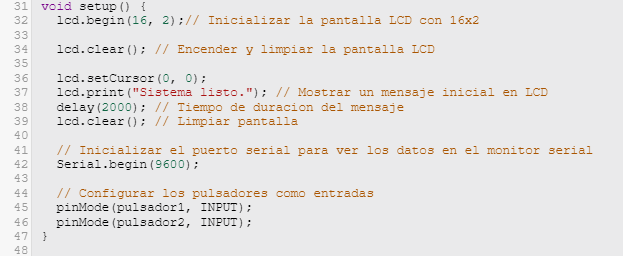
|  |  |
| --- | --- |
| **Pantalla LCD 16x2** | **Potenciómetro** |
| Terminal 1 | GND |
| Wiper | VD |
| Terminal 2 | 5V |

**Algoritmo implementado**

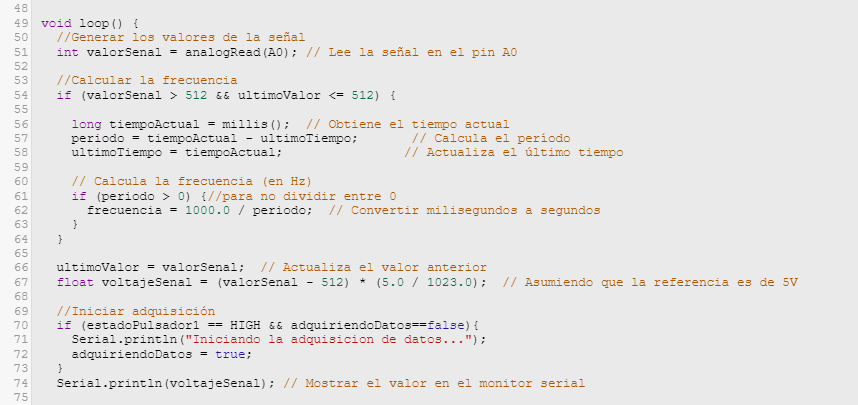
* Declaración de variables:

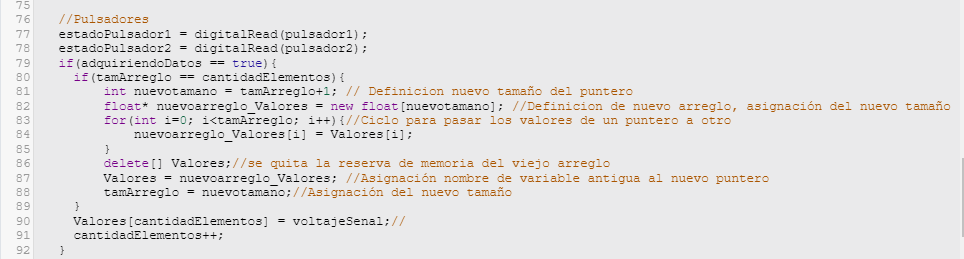
****

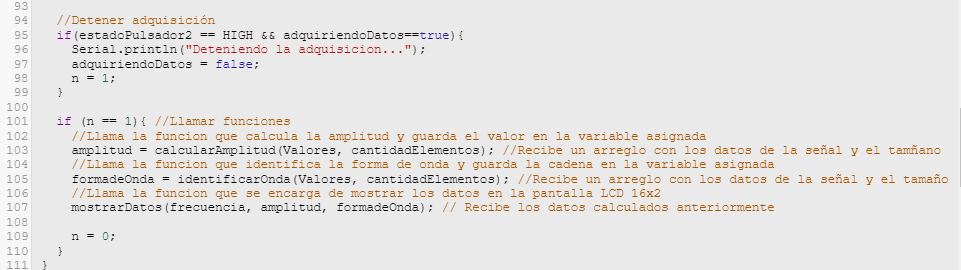
* void setup():

****

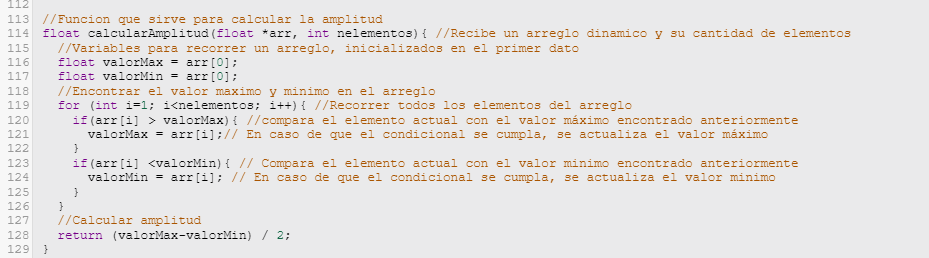
* void loop():

****

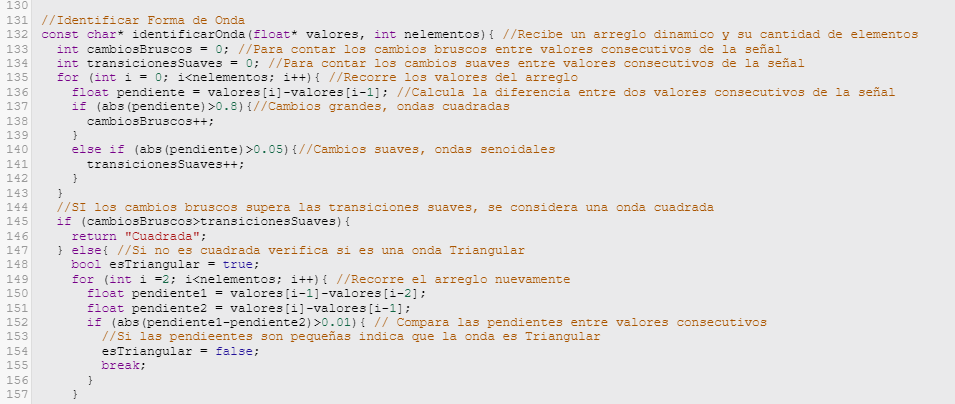
****

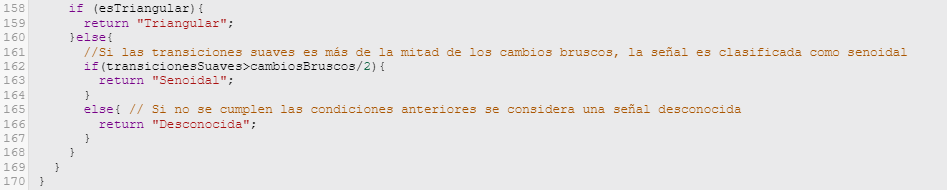
****

* Función para calcular la amplitud

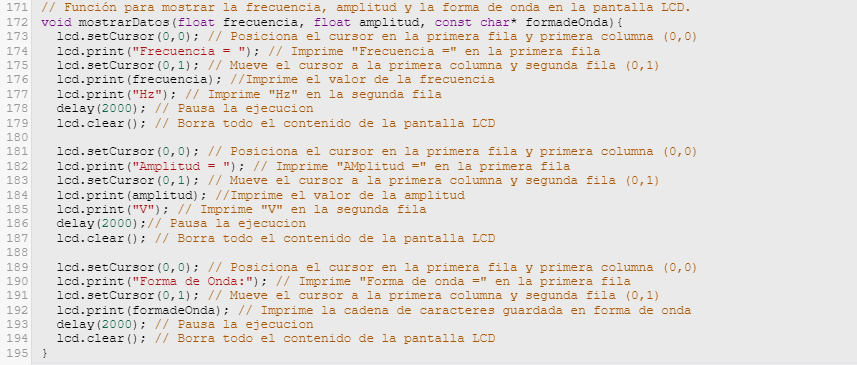
****

* Función para identificar la forma de onda

****

****

* Función para mostrar información en pantalla LDC 16x2

****

1. **Problemas afrontados durante el desarrollo de la solución**

* Tinkercad es una plataforma útil para simulaciones electrónicas, pero presenta varios problemas en su ejecución. Un ejemplo notable ocurre cuando se reciben grandes cantidades de datos en poco tiempo, lo que puede hacer que el sistema colapse o ralentice significativamente su rendimiento. Esto limita la capacidad de realizar análisis en tiempo real o manejar señales con alta frecuencia de muestreo.
* Sabemos que configurando el generador de señales podemos darle valores a la señal acerca de los datos que queremos que se genere. Cuando le colocamos un valor máximo de amplitud sucede que los valores generados por la señal no alcanzan ese valor máximo, solamente se aproxima con un valor menos al indicado. Por tanto, al momento de calcular la amplitud de la señal generada, la respuesta no es igual al valor configurado en el generador.
* Intentamos diversas estrategias para calcular con precisión el valor de la frecuencia. El objetivo principal era que el resultado coincidiera o se aproximara al valor configurado en el generador de señales, basándonos únicamente en los datos proporcionados por este. Para calcular la frecuencia en Hertz, necesitábamos conocer el tiempo de muestreo en segundos, pero no fue posible obtenerlo directamente, ya que los datos obtenidos no nos permitían calcularlo con precisión. Como última opción, decidimos utilizar un tiempo de muestreo estimado para hacer una suposición razonable, calcular el periodo y, a partir de este, determinar la frecuencia.
* Cuando se presionan los pulsadores para comenzar/detener la adquisición, se presentan datos irregulares de los que nos entrega el generador de señales y las mismas variaciones se representan en la gráfica. En algunos casos ocasiona que el entorno se detenga.
* Para grandes variaciones en la frecuencia se ocasiona la detención del entorno en cualquier instante de tiempo.
* En algunos datos no específicos, la generación de señales senoidales se comporta con señal cuadrada, puesto que, se presentan datos repetitivos en los valores máximos y mínimos extremos entregados por el generador de señales y esto se hace evidente en la gráfica.

**6. Consideraciones para la implementación**

* Verifica la precisión de las mediciones de frecuencia y amplitud, ajustando los cálculos según sea necesario. Puesto que, el entorno utilizado presenta problemas para valores indiscriminadamente.
* Asegúrate de que el sistema maneje adecuadamente situaciones inesperadas, como señales fuera del rango esperado (gran número de datos) o fallos en la lectura de los datos.